UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

HASTUS I :
A MATHEMATICAL PROGRAMMING APPROACH TO THE BUS DRIVER SCHEDULING PROBLEM

by

Réjean Lessard, Jean-Marc Rousseau
and Daniel Dupuis*

*Commission des Transports de la Communauté
Urbaine de Québec (C.T.C.U.Q.)

Université de Montréal
Centre de recherche sur les transports - Publication #173
Juin 1980

Dépôt légal - 3e trimestre 1980

ABSTRACT

In this paper, we first present a mathematical programming formulation of the bus drivers scheduling problem in a transit company. Because in general this problem is too large, we introduce a relaxation of the problem and describe a solution strategy. The implementation and results obtained in Québec City are briefly reviewed.

1. The problem

The bus driver scheduling (BDS) problem in a transit company involves establishing at minimum cost for each day of the week, a list of workdays which assign a driver to each bus in the timetable and respect all clauses of the union contract. In the approach discussed here it is assumed that the bus schedule is known and that once the list of feasible workdays is established, the problem is solved. In fact, in most North-American companies, the assignment of workdays to drivers is carried out by the drivers themselves and this selection is done on a seniority basis.

The difficulty of the problem arises directly from the kind of service that a transit company must offer and the travel patterns of the population. Fig. 1 illustrates the service level by time of day for Québec City.

We note that the number of vehicles in service may be much greater at peak hours than at off-peak hours. This obviously necessitates either part time drivers, or split-shift workdays for full time drivers, or both. In most companies, unions are refusing or severely restricting the part-time driver solution. Several rules have then appeared defining legal split-shift workdays and working conditions which limit the number and/or compensate the drivers for less desirable workdays. These working conditions are described in more detail in several papers (Blais, 1976, 1980; Sharp, 1975, R.A.T.P., 1979).

We introduce here the basic terminology and some related rules which characterize the problem.

> A block is the itinerary of a vehicle between its departure from and its return to the garage. It includes all deadhead time required to take the bus in and out of the garage and to and from the route(s) it services. There are generally short blocks to cover the peak periods and long blocks for the basic service.

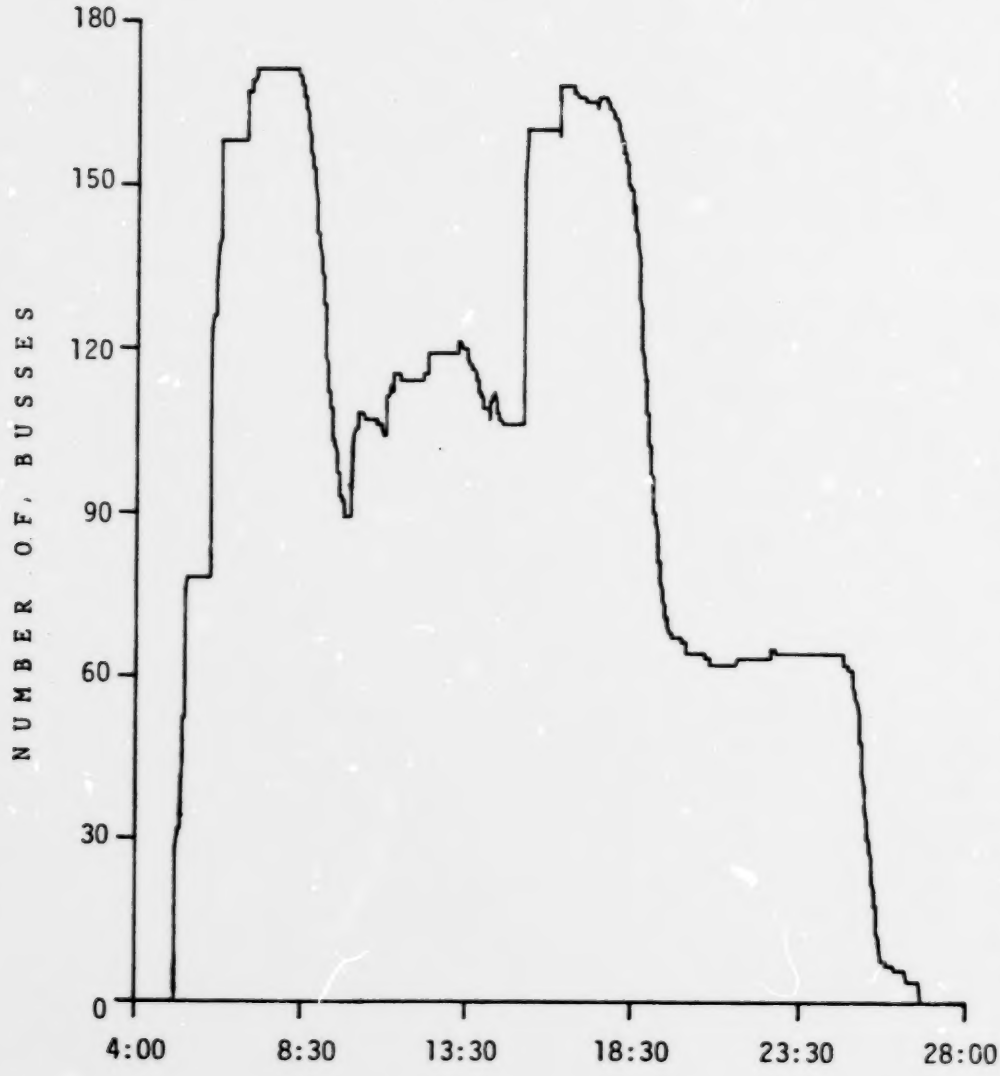


Figure 1 - **Level of service by time of day**

Relief times are the times corresponding to points on the route where a change of drivers is possible. In general the number of such points is small.

A workday is a daily assignment consisting of one or more pieces of work, which must satisfy the union contract rules.

A piece of work (or piece) refers to the period of time during which a driver works continuously with the same vehicle without a break. Generally the number of pieces of work in a workday is limited (2, 3 or 4) and the pieces must have a minimum duration (2-3 hours).

A tripper is a small piece of work which is normally done in overtime by regular drivers or assigned to stand-by drivers. In general the companies hope for a few or no trippers. By extension and simplicity of notation we consider that a tripper is a workday. However we assume that there is no explicit upper bound on the number of trippers thus insuring the existence of a feasible schedule. Moreover, a high penalty is imposed on trippers.

A block partition is a set of pieces of work which covers exactly the block.

The BDS problem has been described in detail and several approximate solution methods have been proposed. The Proceedings from the two workshops on the BDS problem provide an excellent set of references (Preprints,1975; Proceedings,1980).

In the next section we present a general mathematical programming formulation of the BDS problem. Because in general this problem is too large we introduce a relaxation of the problem and present a solution approach. The application of the system in Québec City (250 buses) is briefly reviewed. In another paper (Carraresi, Gallo, Rousseau, 1980) other alternative solution techniques are explored. The notation used in this paper is similar to the one used in the later paper. We borrow heavily from that paper in the presentation of the model.

2. The model

For simplicity of presentation, we now assume that there are at most two pieces of work in a workday. The extension to three or more pieces of work is done later on; in fact in Québec, we use up to three pieces.

The notation for the model is first introduced. By a pair (ij) we denote a piece of work starting at time i and ending at time j. Only feasible pairs (ij) are considered, that is pairs such that both i and j correspond to either a starting time, ending time or relief time in a given block. Note that (ij) could be feasible relative to several blocks. In the first part of this paper however we assume that (ij) is feasible relative to only one block (this could easily be done by small perturbations). For practical reasons, we also include in the feasible set of pieces of work the null pieces (00).

A quadruplet (ijkh) denotes a workday made up with the feasible pairs (ij) and (kh). Only the workdays (ijkh) which are feasible within the union contract

and the company regulations are considered. If (ij) or (kh) is a null piece the workday is either without a break or corresponds to a tripper.

In addition, we define:

L : the number of blocks

K : the number of distinct pieces of work

I : the set of feasible quadruplets (ijkh)

$I(ij) \subset I$: the set of feasible quadruplets (mnkh) where one of the pieces is (ij); it contains piece (00)

T_ℓ : the set of all times which are either relief times, the starting time or the ending time for block ℓ

x_{ijkh} : a binary variable taking value 1 if and only if a driver is assigned to workday (ijkh)

x : the vector with component x_{ijkh}, $(ijkh) \in I$

y_{ij} : a binary variable taking value 1 if the piece (ij) is used to be part of a driver workday. If $y_{ij}=1$, the piece (ij) has been chosen as part of the partition of the given block relative to which it has been defined.

y : the vector component y_{ij} for <u>all</u> feasible (ij)

c_{ijkh} : the cost of a workday composed of the piece (ij) and the piece (kh) according to the union contract.

The problem can now be formulated as follows:

$$(2.1) \qquad \text{Min} \sum_{I} c_{ijkh}\, x_{ijkh}$$

$$\text{s.t.} \quad \text{i)} \quad \sum_{I(ij)} x_{mnkh} - y_{ij} = 0 \qquad \text{for all (ij) except (00)}$$

$$\text{ii)} \quad D\,x \geq d$$

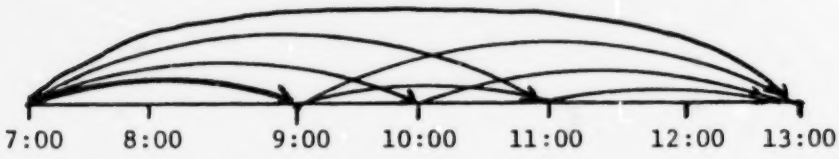
$$\text{iii)} \quad \sum_{i \in T_\ell} y_{ik} - \sum_{j \in T_\ell} y_{kj} = b_k^\ell \qquad \text{for all } k \in T_\ell, \text{ for all } \ell$$

$$\text{where } b_k^\ell = \begin{cases} -1 & \text{if k is the starting time of block } \ell \\ +1 & \text{if k is the ending time of block } \ell \\ 0 & \text{otherwise} \end{cases}$$

$$\text{iv)} \quad x_{ijkh} = 0,1 \ , \quad y_{ij} = 0,1$$

The constraints (iii) correspond to the flow formulation used to partition each block ℓ into pieces of work. Fig 2. illustrates the concept. The feasible

pieces of work (we are assuming here a minimum lenght of two hours) are represented by arcs and listed in the figure. We have assumed for simplicity that a relief point exists every hour in this example. The indices k in the constraint formulation correspond to points where a change of drivers is feasible and the constraints ensure that a flow of one goes from the origin of the block (7:00) to its end (13:00), using arcs (ij) corresponding to feasible pieces. Constraints (iii) define a flow on an uncapacitated network.



The feasible pieces are

7 - 9	7 - 11	9 - 11	10 - 13
7 - 10	7 - 13	9 - 13	11 - 13

Figure 2. The flow formulation for the partition of a block

With constraint (i) we ensure that any feasible piece (ij) used to partition a block will be used in a workday of type x_{mnkh} ($x_{mnkh} \epsilon I(ij)$).

In fact, given the values of the y_{ij}, i.e. the partition of the blocks into pieces of work, constraint (i) with the objective function can be reformulated into a maximum weight matching problem (described in section 6).

Constraints of type (ii) refer to other constraints of the union contract. Examples of such constraints are:

- a minimum or maximum number of workdays without a break or with a limited break
- a limit on the number of drivers
- a limit on the average length of a workday
- etc.

Unless the problem is small (i.e. the number of blocks is small and thus the number of pieces and workdays is limited), this formulation seems impractical. Given a medium size transit network as in Québec City we can easily generate over ten thousand pieces of work and five million workday variables without considering the flow variables and the difficulty of determining integer solutions for x_{ijkh}. In fact this formulation is nearly equivalent to the set

covering formulation found in Heurgon (1972,1975). The set covering formulation was used in Paris to solve the problem one route at a time (drivers were not allowed to change route). However, the formulation (2.1) seems to be more amenable to a solution strategy than can handle very large problems.

3. Solution strategy

The chosen solution strategy is to use the obvious decomposition of the problem into the generation of a partition of each block into pieces of work (constraint (iii)) and the matching problem (constraint (i)) to form workdays. It has to be ensured that constraints (ii) are also satisfied. Three main steps compose this strategy:

Step 1: Using a relaxation of the whole problem we generate a partition of each block into feasible pieces of work that will respect as much as possible the constraint set (ii).

Step 2: Using an assignment algorithm and a heuristic procedure to split the pieces of work into two categories, we solve heuristically the matching problem to obtain a solution to the BDS problem. (Recently, a very fast matching algorithm has been developed by Derigs (Bodin, 1980) and it is planned to replace the assignment algorithm by this matching algorithm).

Step 3: Using a set of heuristic techniques, the solution previously obtained is improved, and it is made sure that constraint set (ii) is respected.

The solution found in all test cases in Québec City and Montréal were either better or comparable to manual solutions. The process has been implemented in Québec City since March 1979 and is currently being developed into a package for the Montréal transit authority. Each of the steps are described in more detail, in the following sections and results from the use of HASTUS I in Québec City are reported.

4. A relaxation of the model: the HASTUS-macro approach

Firstly the integrality of the x variables is relaxed since several methods exist to derive reasonably good integer solutions when a continuous solution has been found. Secondly we assume that the starting times, relief times and ending times for the blocks may only occur at predetermined times $t \in T$, for example every 15 or 30 minutes. In the latter case this means that all bus blocks are approximated to the nearest half hour and relief points are possible at some of or at each half hour period. More complicated schemes could also be devised; for example one could use different time periods for peak and off-peak times. This relaxation of the problem considerably reduces the number of possible pieces of work (ij); however it is important to note that *a piece (ij) may now be feasible relative to several blocks* which also means that x_{ijkh} *may be greater than* 1. All i, j, k, h are now in T.

Moreover, the problem is further relaxed by requiring that the workdays selected, be sufficient to cover the total requirement of drivers per time period (i.e. from one predetermined time to the next), instead of requiring that they exactly cover all the blocks individually. Using the same notation this relaxed problem can be written as follows:

$$(4.1) \qquad \text{Min} \sum_{I} c_{ijkh} x_{ijkh}$$

$$\text{s.t. i)} \quad \sum_{I(t)} x_{ijkh} \geq N_t \qquad \text{for all } t \epsilon T$$

$$\text{ii)} \quad Dx \geq d$$

$$\text{iii)} \quad \sum_{I(pq)} x_{ijkh} \geq Q_{pq} \qquad \text{for all (pq) such that a small block exist from p to q}$$

$$\text{iv)} \quad \sum_{L(t)} x_{ijkh} \geq K_t \qquad \text{for all } t \epsilon T$$

$$\text{v)} \quad x_{ijkh} \geq 0 \quad \text{(integrality is relaxed), } i,j,k,h \epsilon T$$

where:

T : the set of predetermined time which could be relief time, starting and ending times of blocks

$I(t)$: the set of workdays $(ijkh) \epsilon I$ such that $i \leq t < j$ or $k \leq t < h$

$L(t)$: the set of workdays $(ijkh) \epsilon I$ with a piece starting at time t ($i=t$ or $k=t$)

N_t : the number of buses in operation during time period starting at t

Q_{pq} : the number of blocks from p to q

K_t : the number of blocks starting at t

The set of constraints (i) ensures that during all periods of the day the number of drivers working is greater than or equal to the number of vehicles in circulation. Constraint (ii) refers to union contract constraints as previously described. In constraint (iii), the number of pieces of work from p to q is at least as large as Q_{pq} the number of small blocks from p to q. A small block is defined by the user as a block that cannot be partitioned and should be allocated as one piece of work to a driver. This generally corresponds to blocks with a duration less than twice the minimum duration of a piece of work.

In constraint (iv) the number of pieces of work beginning at t must at least be equal to K_t the number of blocks starting at t. Finally, x_{ijkh} is a continuous variable that can take any positive value in this relaxation. However for x_{ijkh} to exist there must be a piece (ij) and a piece (kh) each feasible with respect to at least one block.

The HASTUS-macro approach is independently described in several other papers (Blais, 1976, 1980; Rousseau, 1978) and has been used on several occasions to analyse modifications to the drivers' union contract. A package for the utilisation of HASTUS-macro has also been developed (Blais, 1978) and implemented both in Québec City and Montréal and was extensively used by these companies during their last union contract negotiations.

5. Partitioning the blocks

In the present context however, the HASTUS-macro approach is used to help generate a first feasible solution as close as possible to the lower bound it indicates. This is done first by generating an initial block partition that uses similar types of pieces of work and in approximately the same number as indicated by HASTUS-macro. Until recently, this was achieved by first generating for each block a set of partitions made up of pieces used in work-days corresponding to positive variables x_{ijkh} in the optimal solution of (4.1). A linear programming algorithm was then set up to choose one of the partitions generated for each block in order that the pieces thus chose correspond as closely as possible to the solution of the HASTUS-macro problem (4.1). However, we recently adapted our work with Gallo, Carraresi and Davini (Davini, 1980) and will shortly implement in Québec City the technique described here which achieves the same purpose more efficiently. The following problem is considered.

$$(5.1) \qquad \text{Min } z = \sum_{(ij)} \Big(\sum_{I(ij)} \bar{x}_{mnkh} - \sum_{\ell} y_{ij}^{\ell} \Big)^2 + \sum_{\ell} \sum_{(ij)} d_{ij}^{\ell} \, y_{ij}^{\ell}$$

where

$$\sum_{i \in T_\ell} y_{ik}^{\ell} - \sum_{j \in T_\ell} y_{kj}^{\ell} = b_k^{\ell} \qquad \text{for all } k \in T_\ell \text{ , for all } \ell$$

$$y_{ij}^{\ell} = 0,1 \; ; \quad i,j \in T \; , \quad T_\ell \subset T$$

$\bar{x}_{mnkh}$ correspond to the optimal continuous solution of (4.1)

y_{ij}^{ℓ} is a binary variable taking value 1 if piece (ij) defined on times periods $(i,j \in T)$ and feasible relative to block ℓ is used in the partition of block ℓ

d_{ij}^{ℓ} is a penalty associated with the use of the piece (ij) on block ℓ relative to the real relief, starting and ending times on block ℓ.

As in problem 2.1, the constraints correspond to the formulation of an uncapacitated flow problem.

This problem can easily be solved with an heuristic procedure. In fact, note that if we consider all the variables not associated with block r fixed, (i.e. y_{ij}^{ℓ}, $\ell \neq r$), the objective function is reduced as follows:

$$\sum_{(ij)} \Big(\sum_{I(ij)} \bar{x}_{mnkh} - \sum_{\ell \neq r} y_{ij}^{\ell} - y_{ij}^{r} \Big)^2 + \sum_{\ell \neq r} \sum_{(ij)} d_{ij}^{\ell} \, y_{ij}^{\ell} + \sum_{(ij)} d_{ij}^{r} \, y_{ij}^{r}$$

$$= \sum_{(ij)} \Big(\Big(\sum_{I(ij)} \bar{x}_{mnkh} - \sum_{\ell \neq r} y_{ij}^{\ell} \Big)^2 - 2 y_{ij}^{r} \Big(\sum_{I(ij)} \bar{x}_{mnkh} - \sum_{\ell \neq r} y_{ij}^{\ell} \Big) + (y_{ij}^{r})^2 \Big)$$

$$+ \sum_{\ell \neq r} \sum_{(ij)} d_{ij}^{\ell} \, y_{ij}^{\ell} + \sum_{(ij)} d_{ij}^{r} \, y_{ij}^{r} \; .$$

Because $y^r_{ij} = 0,1 \quad (y^r_{ij})^2 = y^r_{ij}$, the previous equation can be written as

$$D + \sum_{(ij)} c^r_{ij}\, y^r_j$$

where $$D = \sum_{(ij)} \Big(\sum_{I(ij)} \bar{x}_{mnkh} - \sum_{\ell \neq r} y^\ell_{ij} \Big)^2 + \sum_{\ell \neq r} \sum_{(ij)} d^\ell_{ij}\, y^\ell_{ij}$$

and $$c^r_{ij} = -2\Big(\sum_{I(ij)} \bar{x}_{mnkh} - \sum_{\ell \neq r} y^\ell_{ij} \Big) + 1 + d^r_{ij}$$

and we can define and solve a shortest path problem for block r defined as:

$$P_r: \quad \text{Min} \sum_{(ij)} c_{ij}\, y^r_{ij}$$

$$\sum_{i \in T_r} y^r_{ik} - \sum_{j \in T_r} y^r_{kj} = b^r_k \qquad \text{for all } k \in T_r$$

$$y^r_{ij} = 0,1 \ .$$

The suboptimal algorithm to solve (5.1) can now be summarized as follows:

1. a) Take any feasible solution y^ℓ_{ij} and evaluate z_o the corresponding value of the objective function
   b) Set $k \leftarrow 1$
2. a) Solve successively P_r for r=1...L note $\bar{y}_k$ the solution attained
   b) Evaluate z_k the objective function attained for $y=\bar{y}_k$.
3. a) If $z_k = z_{k-1}$ stop
   b) $k \leftarrow k+1$ go to 2.

When this algorithm stops, we have a partition of each block into pieces of work defined on periods, closely related to the HASTUS-macro solution. Actual pieces defined on real starting, relief or ending times for the blocks are then cut to correspond as closely as possible to the pieces defined on the periods. We define at this point the set V of feasible pieces of work on real times obtained by this process. The next step consists in building up a first feasible solution.

6. The matching problem

A maximum weight matching problem can be set up to generate the best set of workdays with a minimum number of trippers. This problem can be defined as follows:

$$(6.1) \qquad \text{Max} \sum_{I'} \bar{c}_{ijkh} x_{ijkh}$$

$$\sum_{I'(ij)} x_{mnkh} \leq 1 \quad . \qquad (ij)\epsilon V$$

$$x_{ijkh} = 0,1$$

where $\bar{c}_{ijkh} = M - c_{ijkh}$

V : the set of feasible pieces defined on real times resulting from the partitioning of the blocks

I' : the set of feasible workdays using pieces from V

I'(ij) : the set of feasible workdays $(mnkh)\epsilon I'$ where one of the piece is (ij)

M : a large number; it corresponds to the relative penalty associated with a tripper.

Note that contrary to problem (2.1) only the x_{ijkh} which are feasible and use pieces of work previously generated by the partition algorithm are generated. A matching code can be used for the solution of this problem. However, with the currently available code, and the size of the problem generated (500 nodes, 10 000 arcs), it tends to use up a great amount of computer time. Until a more rapid matching code become available, we approximate the problem (6.1) by an assignment type problem that we solve with a minimum cost flow algorithm. (As discussed previously, we may use a new matching code soon.)

To do this, the set V of pieces of work is first split into two subsets so that there are only very few matching possibilities within each subset and a maximum of matching possibilities between the two subsets. This objective is achieved by following the indications of HASTUS-macro. We put in the first set A the pieces which occur either in the morning or the evening and in set P the remaining afternoon pieces. An afternoon piece in the macro is either the second piece of a workday connected with a morning piece or the first piece of a workday connected with an evening piece. The dummy piece (00) is added to both sets. The cost c_{ijkh} corresponds to the actual cost of the workday (ijkh). If either (ij) or (kh) is the dummy piece (00) c_{ijkh} is the cost of the tripper or the workday without break. The flow problem corresponding to problem (6.1) is described below and with RNET (Grigoriadis, 1979) we are able to solve our problem (500 nodes, 10 000 arcs) in about 15 sec CPU on a CDC 173.

The assignment problem can be written as

$$(6.2) \qquad \text{Min} \sum_{I'} c_{ijkh} x_{ijkh}$$

$$\sum_{(kh)\epsilon P} (x_{ijkh} + x_{khij}) = 1 \text{ for all } (ij)\epsilon A - (00)$$

$$\sum_{(kh)\epsilon A} (x_{ijkh} + x_{khij}) = 1 \text{ for all } (ij)\epsilon P - (00)$$

The solution obtained uses only feasible workdays; however constraints (ii) of (2.1) may not be respected and several trippers may remain. The heuristic described in the following section is designed to further eliminate the trippers (between 10 and 20 at this step according to our experience in Québec City) and restore feasibility (very slightly violated).

7. A marginal improvement heuristic

The main process of this heuristic involve marginally replacing each partition of each block by an alternate partition. This is achieved as follows:

Step 1: For each block generate the set B_ℓ of all (if not too many) partitions that use only pieces (ij) corresponding to a positive $\bar{x}_{ijkh}$ or $\bar{x}_{khij}$ in the optimal solution of HASTUS-macro (4.1). If insufficient partitions are generated pieces corresponding to null $\bar{x}_{ijkh}$ with a small reduced cost may be used. (See Blais, 1976, for more details.).

Step 2: For each block $\ell=1,\ldots,L$

a) take out first the partition p_o of block ℓ used in the matching problem (either 6.1 or 6.2);
b) consider the resulting set of trippers R_ℓ (composed of trippers in the preceding matching solution and pieces that were matched to pieces of the partition p_o used for block ℓ);
c) choose the partition p_k of B_ℓ, that matched with the trippers of set R_ℓ produces the least cost solution (trippers being highly penalized) which improve feasibility if violated. Replace p_o by p_k and update the matching solution accordingly (p_k may equal p_o).

Step 3: If the solution has improved (cost is reduced or feasibility improved) after considering alternatively each block, go back to step 2. If not, resolve the matching problem (6.1 or 6.2) and stop.

If after these steps a satisfactory solution is not obtained, the solution may be perturbated in different ways to try to achieve a better solution by re-applying Steps 2 and 3 of the heuristic. For example, we arbitrarily increase the number of trippers in the matching solution (by removing a certain number of matches) and reapply the heuristic.

This perturbation applied repeatedly have proved useful to generate solutions with no trippers. In practice however, the CTCUQ is generally satisfied with the first solution produced by the heuristic which may have from 3 to 5 remaining trippers.

At this stage, we could also use any other marginal improvement heuristics in the literature.

8. Variants of the algorithm

8.1. Algorithm modifications for workdays with three pieces of work

In Québec City, workdays with three pieces are permitted and compose in general about ten percent of all workdays. The adaptation of the general strategy described is however straightforward and heuristic in nature. The adaptation of the general formulation (2.1) is direct; variables x_{ijkhmn} are created for such feasible workdays. For the HASTUS-macro formulation, the same comment apply: it is necessary however to limit the number of such variables created, considering only the most probable location in the time table for such workdays. After the partition of the blocks and before the matching problem, it is necessary to pre-match two of the pieces of any three piece workday that emerges from the HASTUS-macro solution. These pre-matched pieces are considered as one piece in the matching problem (6.1 or 6.2). In the marginal improvement heuristic, it may be possible to generate additional three pieces workdays to reduce the number of trippers; such a routine exists in the HASTUS program implemented in Québec City.

8.2. Algorithm modification for workdays without a break

The presence of (and in some cases the necessity for) a certain number of workdays without a break in the solution may considerably reduce the flexibility of the problem and the HASTUS-macro solution may not be as good once these workdays are taken out of the schedule.

We have found it useful to proceed as follows:

Step 1: Use HASTUS-macro on the whole problem.

Step 2: Partition the blocks.

Step 3: Remove from the blocks the pieces corresponding to workdays without a break (make sure there are enough).

Step 4: Use HASTUS-macro on the reduced problem.

Step 5: Partition the blocks.

Step 6: Match the pieces.

Step 7: Heuristically improve the solution.

9. Results and conclusion

This system has been in operation at the Québec City transit authority (CTCUQ) since March 1979. After a period of test it has been used to generate the assignment of drivers for all schedules (week-days and week-ends). Table 1 shows a continuing reduction of the premium paid by the company since the introduction of HASTUS. Even if HASTUS is still more costly for week-end assignment a total saving of 0,9% which represent an annual saving of $125 000 was achieved. This represents 16% of the premiums (which represent the total potential for savings). The CTCUQ is using the system on an IBM 370/148;

it takes 45 min of CPU time. It has also developed a series of printouts to be used directly by the drivers to sign for their assignments. Other reports are also used for administrative purposes.

Note that the system is used even if a sophisticated package is not available. A computer analyst is responsible for the runs of this system and report the results to the scheduler. Occasionnally, several runs are necessary but most of the time one run is enough. The CTCUQ has been very satisfied with this system. Following these results, the Montréal transit authority (CTCUM) (2 000 buses) has decided to adopt this approach. However, for this project a more sophisticated package is currently under development. This package will include several interactive routines to let the schedulers specify additional constraints and modify the solution produced. Implementation is scheduled to start in January 1981 and several reports are planned. Other researches have also been undertaken to study alternative mathematical programming approaches which could improve further the quality of the solution produced. (Carraresi, 1980).

	Manual solution	HASTUS solution		
	Oct 79	Dec 79	March 80	June 80
Week-days	6,03%	5,51%	5,38%	4,79%
Saturday	3,46%	4,65%	4,45%	3,81%
Sunday	3,70%	5,25%	5,45%	4,60%
Weekly average	5,55%	5,40%	5,28%	4,66%

Table 1 - Premium paid in percentage of total salary

REFERENCES

BLAIS, J.-Y *et al.* (1976)."The Problem of Assigning Drivers to Bus Routes in an Urban Transit System". Publication #44, Centre de recherche sur les transports, Université de Montréal.

BLAIS, J.-Y. (1978). "User Manual for HASTUS-macro". Publication #93, Centre de recherche sur les transports, Université de Montréal.

BLAIS, J.-Y. and ROUSSEAU, J.-M. (1980) "HASTUS: An Evaluation Model for Drivers' Union Contract Negotiation in Transit Companies". Publication #163, Centre de recherche sur les transports, Université de Montréal.

BODIN,L. (1980). Private communication.

CARRARESI, P., GALLO, G., ROUSSEAU, J.-M. (1980). "A Decomposition Approach to Large Scale Bus Drivers Scheduling". Presented at Euro IV Conference, Cambridge, July 1980. Publication #172, Centre de recherche sur les transports, Université de Montréal.

DAVINI, A. (1980). "Il problema della formazione dei 'turni-lavoro' per il personale viaggeante di una azianda publica di transporti urbani". University of Pisa, thesis.

GREGORIADIS, M., HSU, T. (1979). "RNET, The Rutgers Minimum Cost Network Flow Subroutines".User Documentation, Version 3.5. Department of Computer Science, Rutgers University, New Brunswick, New Jersey, Sept. 1979

HEURGON, E. (1972). "Un problème de recouvrement: l'habillage des horaires d'une ligne d'autobus". Revue Française d'Automatique, d'Informatique et de Recherche opérationnelle, Vol. 6, No. 1, pp. 13-29.

HEURGON, E. (1975). "Preparing Duty Rosters for Bus Routes by Computer". Preprints of the Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services, 27-29 April, 1975, Chicago.

Preprints of the Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services, 27-29 April, 1975, Chicago.

Proceedings of the International Workshop on Urban Passenger Vehicle and Crew Scheduling, 15-18 July 1980, Leeds, U.K.

ROUSSEAU, J.-M., DUPUIS, D. and BLAIS, J.-Y. (1979). "Affectation des chauffeurs avec le système HASTUS-macro". Routes et transports, 23, août 1979.

R.A.T.P. (1979). Exact reference unknown. Rapport d'une enquête sur les conditions de travail dans les compagnies de transport en commun européennes.

SHARP, G.P. (1975). "Constraints for Scheduling Operators for Urban Public Transit Systems". Preprints of the Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services. April 27-29, 1975, Chicago, U.S.A.